COP4521 Programming Assignment 2: An Apache Web Access Log Analyzer

Objectives:

- Practice problem solving with Python
- Practice string processing, file operations, and the use of various data structures in Python

Description:

The Apache access log format specifies how entries in a web server's access log are structured. Common formats include the Common Log Format (CLF) and the Combined Log Format. CLF is a standardized text-based format used by many web servers. Each line in the log represents a single request and follows this structure:

```
host_ip ident authuser [date] "request" status bytes
```

An example of a line in a CLF log file is as follows:

```
127.0.0.1 - frank [09/Sep/2025:12:34:56 +0000] "GET /index.html HTTP/1.1" 200 2326
```

Each line includes the following fields

- host_ip: IP address of the client
- ident: RFC 1413 identity of the client (usually -)
- authuser: User ID for HTTP authentication
- [date]: Date and time of the request
- "request": HTTP method, resource, and protocol
- status: HTTP response code (e.g., 200, 404)
- bytes: Size of the response in bytes

In the Combined Log Format, each line includes all the fields from the Common Log Format (CLF), optionally followed by referrer and user-agent information. For this assignment, your web access log analyzer will process a log file using the Combined Log Format; however, you can ignore the referrer and user-agent fields and focus only on the CLF fields. Specifically, the analyzer will output the top 30 most active hosts accessing the website and the top 30 most downloaded resources within a specified date range.

Implementation:

The analyzer should accept one required command-line argument, `logFilePath`, which specifies the path to the web access log file. Additionally, it may accept two optional arguments: `start_date` and `end_date`, both in the format `mm/dd/yyyy`. Below are examples of valid commands to run the analyzer, assuming the program is named `sample_assignment2.py`:

```
<linprog6:289> python3 sample_assignment2.py  ../apache_logs
<linprog6:290> python3 sample_assignment2.py  ../apache_logs  05/20/2015
<linprog6:291> python3 sample_assignment2.py  ../apache_logs  05/17/2015  05/17/2015
<linprog6:292> python3 sample_assignment2.py  ../apache_logs  05/17/2015  05/19/2015
```

If not specified, the default start date is `01/01/0000`, and the default end date is `12/30/9999`. Sample outputs from a reference analyzer are provided as test cases in a separate document.

The analyzer must process each line of the log file, checking whether the access date falls within the specified date range (inclusive of both start and end dates). It should track two metrics within this range:

1. The number of times each host accesses the website.
2. The number of times each resource is downloaded.

A resource should only be considered downloaded when the HTTP request method is **GET**. Requests using other methods (e.g., **POST**) must be ignored.

The analyzer should output:

- The top 30 most active hosts.
- The top 30 most downloaded resources.

If there are fewer than 30 unique hosts or resources, only the actual number should be displayed. In cases where multiple hosts or resources are tied for the 30th place, the analyzer must output all entries that share that rank, which may result in more than 30 being shown.

Due time: September 26, 2025, 11:59pm.

Submission: Name your program lastname_firstinitial_assignment2.py and submit on Canvas.

Grading (60 points total):

- A program gets at most 6 points if the program generates a runtime error.
- Include basic header (template at the course website) for assignment, name your program lastname_firstinitial_assignment2.py (10 points)
- Produce the same output as the test cases provided for the small file (apache_log_small) (20 points)
- Produce the same output as the sample program with another test case for the small file (10 points)
- Produce the same outputs as the test cases provided for the larger file (apache_logs) (10 points)

- Produce the same outputs as the sample program with another test case for the larger file (10 points)
- Extra points: The first person who reports a bug in the sample program (which produced the test cases) will get 3 extra points.

Note:

- The sample analyzer has about 120 lines of code.